Self-reporting on user’s status to minimize interruption and to get attention for interaction

Shreejana Prajapati¹, Yamada Koichi²

¹Information Science and Control Engineering, ²Dept. of Management and Information Systems Science
Nagaoka University of Technology, 1603-1 Kamitomioka-machi, Nagaoka, Niigata, Japan.

Abstract
Impact of interruption on work performance is becoming a great issue in current context of HCI. Various ways of interruption prediction has been studied to minimize interruption. In this paper we are proposing a self-reporting intermission in order to get users’ attention for interaction, without interrupting them. Self-reporting intermission gives users the privilege of choosing their appropriate time to handle interruption. Here we study users’ interruptibility level at the time of self-report. And compare it with users’ interruptibility level at the time of application switching and regular time interval. Also we study how frequently users report themselves as being available for interaction.

Keywords: Self-Report, Interruption, Application switching, Regular Intervals

Introduction
Interruption is a prominently occurring phenomenon in human computer interactions which drives users’ attention away from regular work. In general, users spend very little continuous time on any single task [3], e.g. they spend just about 12 minutes in the currently active task [2, 3] before they move to another task. In short, people keep on interrupting themselves in a certain time interval, even if they do not get any external interruption. This might occur as a result of either positive or negative triggers [1]. Positive triggers occur when work is too easy and negative triggers occur when work is too difficult at their level.

We can see adequate amount of research on interruption and its effects but till date self-initiated intermission has not received enough research attention. So, to present a better way of interruptibility prediction with less negative impact on ongoing work, here we introduce a way of determining frequency of self-report (SR) on intermission. Then we will be comparing it with various other ways of interruptions namely; interruption during application switching (AS) and regular interval (I). While interruption at application switching and at regular time intervals really interrupt users’ work, self-reporting in self-initiated intermission gives users the privilege of choosing their appropriate time to handle interruption. Here we aim to show that self-reporting is the better way of getting users attention.

This paper focuses on answering two questions:
1) How frequently users self-report about their status of being available for interruption?
2) What are users’ responses regarding their own status during interval interruption, application switching and at the time of self-reporting? Based on interruptibility find which one is the best among these three?

Related Work
People spend about 12 minutes in a working sphere before switching to another task [1] as people become habitual to interrupt themselves due to the external
influences [5]. Study on task switching behavior [6] found that 59% of users’ task switches were either self-initiated (40%) or the result of moving on to the next task on a task list (19%). These self-initiated switches were introduced as self-interruption [2, 5, 6]. It is a function of organizational environment and human habits [6]. In organizational culture whenever individuals need help to do their work they are willing to switch from ongoing task in order to get necessary information. Since interruption, whether external or self-initiated both have positive and negative impact on ongoing task, one cannot totally eliminate interruption form daily life.

Existing literature focuses on interruption management, causes of internal and external interruption and its impact on current work. None of the literature has focused on analyzing self-reporting intermission. So here we are presenting an approach of determining the frequency of self-reporting intermission and compare it with other ways of interruption.

Proposed Method
Self-reporting intermission is users’ self-initiation to report on their interruptibility level or self-reporting on being available for interaction. In our research we aim to find users’ self-reporting behavior and show that it is better way of getting users’ attention. Studies in the past [4] showed that application switching might be good approach of getting users attention but every application switching that occurs during computer work is not necessarily switching from one complete task to another. Also it might not indicate users are free for interaction. In order to handle this problem concept of hierarchical task break down [7] has been used but it seems unrealistic to break each and every task in real time working environment. So, we came up with an idea of studying users’ self-reporting time and behavior.

We derive following hypotheses based on our research questions:

**H1:** Users get more distracted by continuous interruptions than by interruptions during application switching

**H2:** Interruptions during application switching do not always indicate users are free for interaction.

**H3:** Self-report will trigger less disruption than interruptions at regular interval and at application switching.

Experiment
To study self-reporting intermission we plan series of experiments consisting of multitasking environment.

We did three different experiments; i) experiment to get users’ interruptibility level at application switching, ii) at regular time interval and iii) users’ self-reporting on their interruptibility level. We consider interruptibility in 4 levels (1. Highly Un-interruptible, 2. Un-interruptible, 3. Interruptible and 4. Not at all). Users’ interruptibility level changes depending on the circumstances. Aim of experiment was to check how users respond to incoming messages during different circumstances. Circumstances here refer to application switching and at regular time interval. Also study whether users report themselves as being free for interaction or not. If yes then how often do they report themselves as being free for interaction?

Users’ Task Scenario
In experiment users were asked to perform two tasks, document reading including summarization and solving cross word puzzle. Once users are ready for experiment they were briefed about their task. In total we perform the experiment with 12 users out of 12 users there were
3 females and 9 males, their age ranges from 20-35. All the users are from technical field with academic qualification ranges from undergraduate to post-graduate research students.

**Users’ Task Detail**

Since we are up to three different experiments with different objectives, we asked the subjects to do experiment in random order. While users were doing with the experiment we recorded user physically and contextually. All the context information such as mouse and keyboard event along with desktop activities were recorded. And it was pre-informed to users that their activities both desktop and physical activities will be recorded during the experiment. Users are asked to do the experiment in their own pace and there was no time limitation for completing the task. During the experiment users get pop up message asking to give their interruption level.

**Result**

As per previous research suggested that application switching might be good time to get users’ attentions for interaction, but here in our experiment we found that not all the application switching indicates that users are free for interaction. In our experiment to get users’ interruptibility during application switching we found that almost 80% of the time when the application switching occurs users considered themselves as in Highly Uninterruptible as shown in fig. The result supports hypothesis H2.

Experimental result from interruption at regular time interval shows that users’ interruptibility level was either uninterruptible or highly uninterruptible almost all the time during the experiment except for two users as shown in fig 2.

In experiment for self-report on interruptibility level we found that whenever users report themselves they were always in interruptible state supporting H3 as shown in fig 3. From the chart we can see that users report themselves at least for 4 times and at max for 12 times during task performance. User 6 and 8 who report themselves 4 times took 1 hour 16 minutes and 1 hour 35 minutes respectively, whereas user 11 who respond 12 times took 1 hour 52 minutes to complete the whole task.
Comparison

We performed three different experiments with predefined tasks and almost all of the users take around one and half hour to two and half hour to complete the given task. When we compare the result from three different experiment (fig1, 2, 3), occurrence of interruptibility level Highly Un-Interruptibility was higher at application switching. Interruptibility level was in between Highly Uninterruptible and Uninterruptible most of the time during interruption at regular time interval and it was always in interruptible state during self-reporting. And in case of interruption at application switching users were in interruptible state in some point but they were hardly in interruptible state at regular time interval supporting H1. These shows that self-reporting might be better among these three.

Discussion

From the experimental data we found that most of the time when an interruption occurs during application switching and at regular time interval users considered themselves as Un-Interruptible. And also self-reporting resulted that users become available for interruption quite often and that didn’t hamper their regular work. We had post experimental questionnaire where all of the users except two have said that appearance of pop-up message was much annoying than reporting by themselves and they prefer reporting themselves for intermission.

Conclusion

Here we propose a self-reporting intermission for getting users’ attention without creating annoyance and anxiety to users. And the distinguishing feature of our work is to overcome the problem of getting users’ attention for interaction without interrupting them.

References: